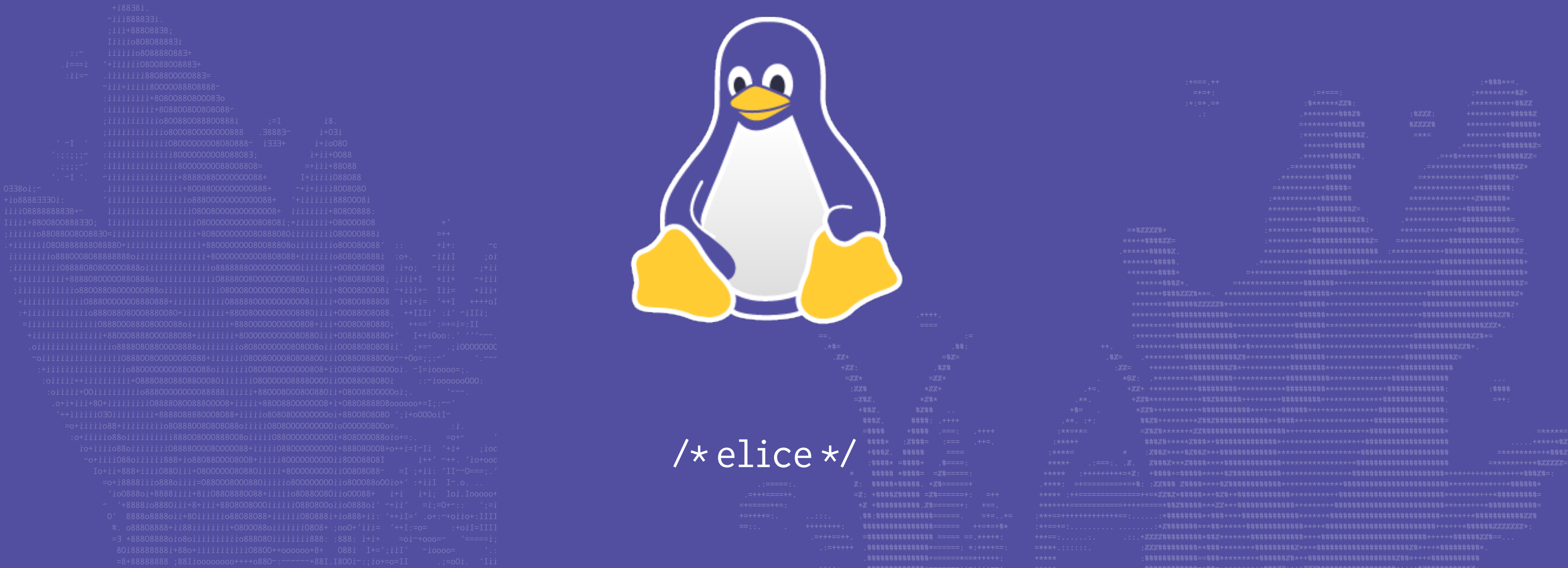


리눅스 기초

알고 쓰는 리눅스 명령어



수강 목표

정규표현식의 **개념**을 이해합니다.

문자열과 관련된 **명령어**를 알아봅니다.

리눅스 기반의 **CLI 환경**에 친해집니다.

목차

1. 정규표현식
2. grep
3. File Redirection
4. Piping commands
5. 마운트

정규표현식

정규 표현식

정규 표현식 (Regular expression)은 **특정한 규칙**을 가진
문자열의 집합을 **표현**하는 데 사용하는 일종의 형식

정규표현식 문법

특정 문자들은 특정한 논리적 관계를 나타내는 기호

이를 제외한 나머지는 일반적인 문자를 뜻함

정규식 메타 문자

메타 문자	의미	메타문자	의미
^	문자열의 시작	\d	숫자
\$	문자열의 끝	\D	숫자가 아닌 것
.	하나의 문자와 대응	\w	알파벳 대소문자, 숫자, 언더스코어(_)
\s	공백문자	\W	/w에 해당하지 않는 글자
\S	공백문자가 아닌 것	\n	개행문자
\	메타 문자를 문자 그대로 사용	\t	탭(tab) 문자

선택 패턴

| 문자를 이용하면 `A | B` 의 패턴으로 매칭 가능

예를 들어 `alice|elice` 라는 정규 표현식은

“alice”와 “elice”라는 문자열을 모두 포함한다.

즉, alice 또는 elice 둘 중 하나라는 뜻

선택 패턴

[] 대괄호 속에 넣은 문자 중에서 하나에 매칭

[0-9] 0부터 9까지의 숫자 하나에 매칭

[A-Z] A부터 Z까지의 알파벳 대문자 하나에 매칭

[] 내에서 ^ 기호는 not의 의미를 가지게 된다.

수량 한정자

수량자	의미	수량자	의미
?	0개 또는 1개	{ n }	n개
*	0개 이상	{n, m}	n개 이상, m개 이하
+	1개 이상	{n, }	n개 이상

그룹 패턴

`()` 괄호는 그룹을 나타냄

그룹은 전체 패턴 내에서 하나로 묶여지는 패턴

한 번 생성한 그룹은 재사용 가능

만들어진 순서로 1번, 2번, 3번으로 참조 가능

예를 들어 `(to)ma\1` 은 'tomato'와 매칭됨.

정규 표현식 사용 예

전화번호는 **일정한 패턴**을 가진 문자열

아래의 조건을 만족하는 전화번호를 찾는 정규표현식은?

- * 전화번호는 010으로 시작한다.
- * 전화번호의 가운데 자리는 네 자리이다.
- * 전화번호의 각 자리에 구분문자가 있을 수도 있고, 없을 수도 있다.

정규 표현식 사용 예

```
^010(\D?\d{4}){2}
```

^

시작 문자는 010

\D?

숫자가 아닌 문자가 0개 이상

\d{4}

숫자가 4개

(\D?\d{4}){2}

앞의 그룹이 2개 존재

grep

grep

grep 명령은 파일 내에서 지정한 패턴이나 문자열을
찾은 후에 그 패턴을 포함하고 있는 모든 행을 출력

grep

```
grep [option] [pattern] [파일명]
```

option

- i : 대소문자를 구분하지 않고 검색한다.
- v : 패턴과 일치하지 않는 행을 출력한다.
- c : 패턴과 일치하는 행의 개수를 출력한다.
- w : 패턴과 단어 단위로 매칭되어야 출력한다.

grep 예시

```
grep elice hello.txt
```

hello.txt 파일에서 'elice'라는 문자열이 들어 있는 행을 모두 출력

```
grep -c elice hello.txt
```

hello.txt 파일에서 'elice'라는 문자열이 들어 있는 행의 수 출력

grep과 정규 표현식

```
grep [0-9] hello.txt
```

hello.txt 파일에서 숫자가 존재하는 행을 모두 출력

```
grep "Elice\.The\.Rabbit" *
```

현재 디렉토리의 모든 파일에서 Elice.The.Rabbit을 찾아 출력

grep과 정규 표현식

```
grep -v ^# somecode
```

somecode 파일에서 #으로 시작하지 않는 행을 모두 출력

```
grep '\<[a-z].*e\>' elice
```

elice 파일에서 소문자로 시작하고 공백을 포함한 여러 문자가 나오며, e로 끝나는 단어가 포함된 행을 모두 출력

File Redirection

File Redirection

File Redirection은 표준 스트림의 흐름을 바꾸어
일반적인 표준 입력 및 출력 그리고 오류를 사용하지 않고
다른 경로인 파일로 재지정하는 것을 뜻합니다.

Standard Stream

일반적으로 표준 입력, 표준 출력, 표준 오류 출력으로 분류
스트림은 문자열로 콘솔에 출력되도록 설정되어 있음

stdin

키보드 입력

stdout

화면 출력

stderr

오류 내용 출력

File Redirection


```
ls > ls.txt
```

> 연산자는 표준 출력을 재지정

```
ls >> ls.txt
```

>> 연산자는 파일이 존재하지 않다면 파일을 생성,
존재한다면 파일내용을 지우지 않고 이어서 작성

File Redirection

표준 오류는 연산자를 사용하지 않으며
파일디스크립터 번호를  앞에 작성해서 사용

0

표준 입력

1

표준 출력

2

표준 에러

File Redirection

```
elice@elice-VirtualBox: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
elice@elice-VirtualBox:~$ cd helloelice > err.txt  
bash: cd: helloelice: 그런 파일이나 디렉터리가 없습니다  
elice@elice-VirtualBox:~$ cat err.txt  
elice@elice-VirtualBox:~$ cd helloelice 2> err.txt  
elice@elice-VirtualBox:~$ cat err.txt  
bash: cd: helloelice: 그런 파일이나 디렉터리가 없습니다  
elice@elice-VirtualBox:~$
```

File Redirection

```
cat < hello.txt > elice.txt
```

1. hello.txt의 내용을 cat 명령어의 입력 스트림으로 전송
2. cat 명령어는 hello.txt 파일의 내용을 출력
3. cat 명령어의 출력 스트림을 elice.txt로 변경
4. cat 명령어의 출력 스트림은 화면이 아닌 elice.txt에 저장

Piping commands

Linux Pipe

Pipe는 둘 이상의 명령어를 묶어 출력의 결과를

다른 명령으로 전환할 수 있는 기능

명령어의 표준 출력을 또 다른 명령어의 표준 입력으로 연결

| 기호를 사용하여 명령어와 명령어를 연결

Linux Pipe

```
head a.txt | grep [0-9]
```

1. head 명령을 실행하여 a.txt의 첫 10줄을 출력
2. 출력된 결과를 | (pipe)를 통해 grep 명령으로 전달
3. 숫자가 포함된 행을 가진 행의 결과가 모두 출력

Linux Pipe

```
elice@elice-VirtualBox:~/문서/log$ head a.txt | grep [0-9]
3 * 3 is 9
20190903
Welcome 2 Elice!
elice@elice-VirtualBox:~/문서/log$
```

pipe를 통해 복잡한 작업을 보다 간단히 수행 가능

Pipe & Redirection

```
head a.txt | grep [0-9] > result.txt
```

File Redirection, Pipe를 조합하여 더욱 다양하고

효율적인 작업을 수행할 수 있다.

Pipe & Redirection

```
ls | grep user01 > output.txt
```

1. ls 명령을 실행 (현재 디렉토리의 디렉토리/파일 명 출력)
2. ls 명령의 결과를 입력 값으로 한 grep 명령어 실행
3. user01이라는 이름을 가진 파일의 결과가 출력
4. grep 명령어의 출력 스트림은 output.txt에 저장

Pipe & Redirection

```
elice@elice-VirtualBox:~/문서/log$ ls | grep user02 > output.txt
elice@elice-VirtualBox:~/문서/log$ cat output.txt
user02_20190807
user02_20190813
user02_20190818
user02_20190830
user02_20190910
user02_20190924
elice@elice-VirtualBox:~/문서/log$
```

output.txt 파일 안에 수행한 명령의 결과가 저장

마운트

mount

mount란 물리적인 저장 장치(보조기억장치)를
디렉토리(또는 폴더)에 연결시켜주는 것을 말함.

mount

윈도우에서는 하드, USB등의 보조기억장치를 연결하면

자동으로 폴더(디렉토리)에 연결 됨.

이것을 PnP(Plug and Play)라고 함.

mount

리눅스의 경우 PnP 기능이 작동하지 않아 직접 연결해야함.

즉, 보조기억장치를 설치했을시 mount 작업을 해야 사용 가능

mount

```
mount [option] [device] [directory]
```

option

- a : /etc/fstab에 명시된 파일 시스템을 마운트 할 때 사용
- t : 파일 시스템의 유형을 지정, 생략할 시 /etc/fstab 파일을 참조
- o : 추가적인 설정을 적용할 때 사용, 다수의 조건을 적용할 때는
coma(,)로 구분

mount

```
remount [device] [directory]
```

mount를 취소하는 명령어

```
df
```

현재 mount 된 디스크 정보 출력

CREDIT

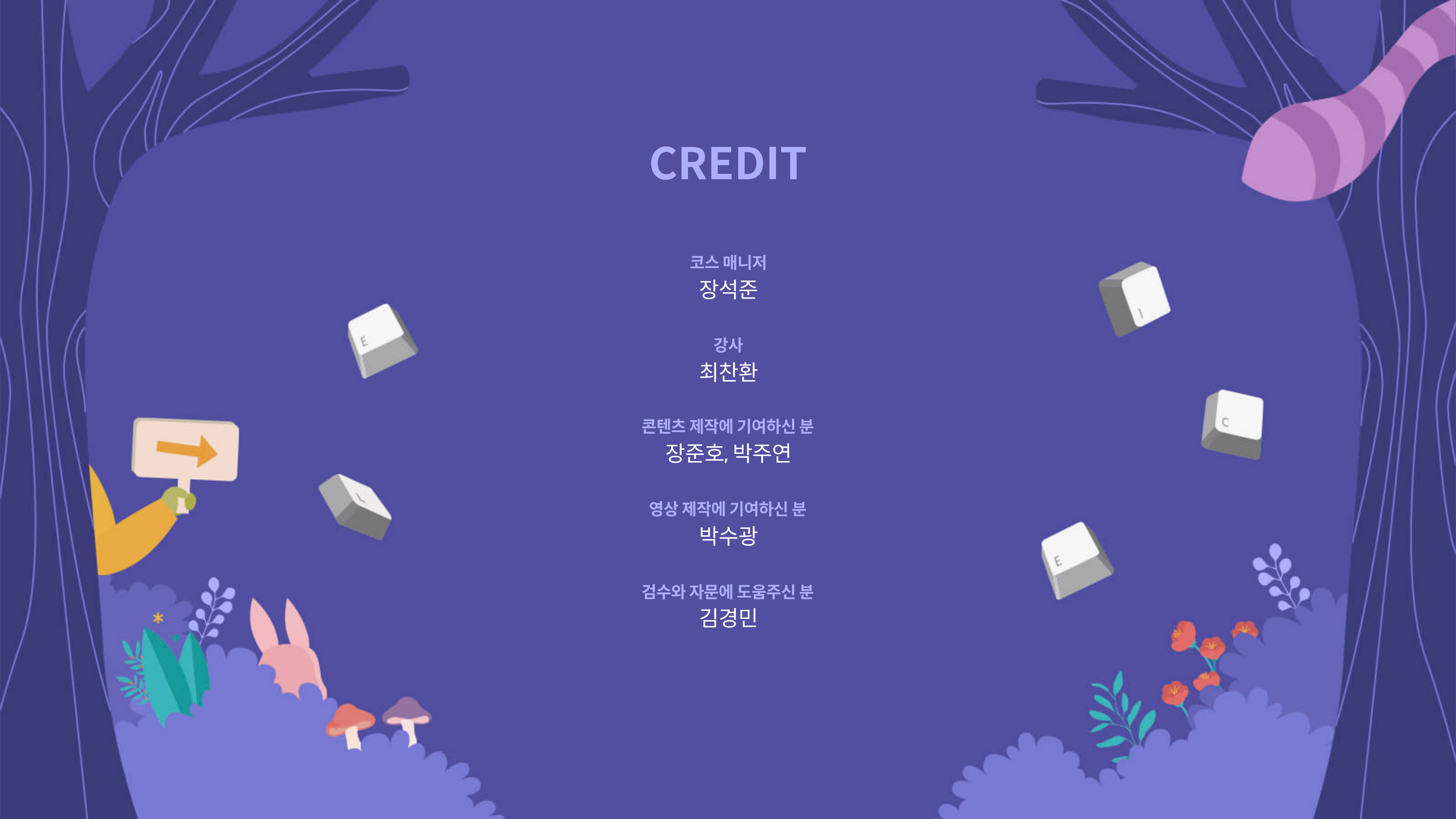
코스 매니저
장석준

강사
최찬환

콘텐츠 제작에 기여하신 분
장준호, 박주연

영상 제작에 기여하신 분
박수광

검수와 자문에 도움주신 분
김경민





The background is a deep purple forest floor. On the left, a yellow hand holds a sign with an orange arrow pointing right. Several white keyboard keys with black characters (E, I, C, L) are floating in the air. In the bottom left, there are pink bunny ears, green leaves, and two mushrooms. In the bottom right, there are red flowers and green leaves. A pink and white striped caterpillar is on the right edge. The central text is in a white, monospace-style font.

`/*elice*/`

contact@elice.io